

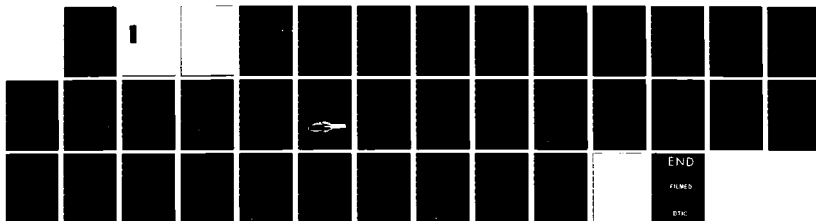
AD-A159 317

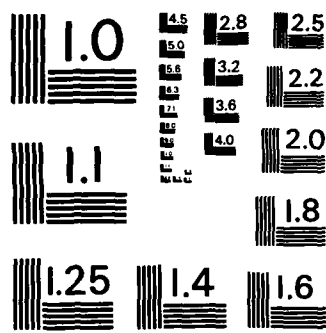
INMESH: AN INTERACTIVE PROGRAM FOR NUMERICAL GRID
GENERATION(U) DAVID M TAYLOR NAVAL SHIP RESEARCH AND
DEVELOPMENT CENTER BETHESDA MD R H COLEMAN AUG 85
DTNSRDC-85/054 F/G 9/2

1/1

UNCLASSIFIED

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

AD-A159 317

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

AD-A159 317

REPORT DOCUMENTATION PAGE

1a REPORT SECURITY CLASSIFICATION UNCLASSIFIED		1b RESTRICTIVE MARKINGS	
2a SECURITY CLASSIFICATION AUTHORITY		3 DISTRIBUTION/AVAILABILITY OF REPORT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.	
2b DECLASSIFICATION/DOWNGRADING SCHEDULE			
4 PERFORMING ORGANIZATION REPORT NUMBER(S) DTNSRDC-85/054		5 MONITORING ORGANIZATION REPORT NUMBER(S)	
6a NAME OF PERFORMING ORGANIZATION David W. Taylor Naval Ship R&D Center	6b OFFICE SYMBOL (if applicable) DTNSRDC	7a NAME OF MONITORING ORGANIZATION	
6c ADDRESS (City, State, and ZIP Code) Bethesda, Maryland 20084-5000		7b ADDRESS (City, State, and ZIP Code) SEP 23 1985	
8a NAME OF FUNDING SPONSORING ORGANIZATION Naval Sea Systems Command	8b OFFICE SYMBOL (if applicable) SEA 05R24	9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER A	
10 SOURCE OF FUNDING NUMBERS			
11 ADDRESS (City, State, and ZIP Code) Department of the Navy Washington, D.C. 20362		PROGRAM ELEMENT NO 61153N	PROJECT NO SR01403
		TASK NO SR0140301	WORK UNIT ACCESSION NO DN 678303
12 TITLE (Include Security Classification) INMESH: AN INTERACTIVE PROGRAM FOR NUMERICAL GRID GENERATION			
13 PERSONAL AUTHOR(S) Roderick M. Coleman			
14a TYPE OF REPORT Final	13b TIME COVERED FROM TO	14 DATE OF REPORT (Year, Month, Day) 1985 August	15 PAGE COUNT 37
16 SUPPLEMENTARY NOTATION			
17a COSATI CODES		18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number)	
GROUP	SUB-GROUP	Numerical Mesh Generation	
		Boundary-Fitted Coordinates	
19 ABSTRACT (Continue on reverse if necessary and identify by block number) INMESH is a computer program for the generation of two-dimensional boundary-fitted coordinate systems. This general program creates a numerical transformation which maps an arbitrary region in physical space onto a regular region with a uniform grid in computational space. The computational region may consist of one or more segments joined together so that any topological complexity in the physical region, such as multiple-connectedness, is suitably resolved. Numerical solutions of partial differential equations may be obtained from this computational field regardless of boundary shape. Emphasis is placed on ease of use and interactive graphics so that a mesh can be generated quickly, viewed interactively, and regenerated immediately with slight alterations until a suitable mesh configuration has been obtained. A brief explanation of the method is given along with a detailed description of the input required to (Continued on reverse side)			
20 DISTRIBUTION AVAILABILITY OF ABSTRACT <input checked="" type="checkbox"/> UNCLASSIFIED UNLIMITED <input type="checkbox"/> SAME AS RPT <input type="checkbox"/> DTIC USERS		21 ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED	
22a NAME OF RESPONSIBLE INDIVIDUAL Roderick M. Coleman		22b TELEPHONE (Include Area Code) (301) 227-1930	22c OFFICE SYMBOL DTNSRDC Code 1843

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

(Block 19 continued)

produce boundary-fitted coordinate systems. Several sample meshes are shown for configurations involving foil-like bodies.

Accession No.	
FILE NO.	
DATE	
Version	
Justification	
Distribution/	
Availability Codes	
Serial and/or	
Special	



UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

TABLE OF CONTENTS

	Page
LIST OF FIGURES.....	iii
LIST OF TABLES.....	iv
ABSTRACT.....	1
ADMINISTRATIVE INFORMATION.....	1
INTRODUCTION.....	1
DESCRIPTION OF THE METHOD.....	3
USE OF THE PROGRAM.....	5
EXAMPLES AND APPLICATIONS.....	7
CONCLUSION.....	15
ACKNOWLEDGMENT.....	15
APPENDIX A - INMESH USER'S GUIDE.....	17
APPENDIX B - DPAC: DISTRIBUTE POINTS ON A CURVE.....	27
REFERENCES.....	29

LIST OF FIGURES

1 - Example 1: C-type Grid for Foil.....	8
2 - Example 1: Grid for Region Shown in Figure 1 Generated without Coordinate Line Control.....	9
3 - Example 2: C-type Grid for Foil with Slot.....	10
4 - Example 2: Close View of Foil with Slot.....	11
5 - Example 3: Grid for Multibody Foil.....	12
6 - Example 3: Close View of the Confluence Region on the Multibody Foil.....	13
7 - Example 4: Grid for Foil in Rectangular Region.....	14
8 - Free-Surface Development for Translating Hydrofoil.....	15

	Page
A.1 - Mesh 1: Computational Region.....	21
A.2 - Mesh 1: Physical Region.....	22
A.3 - Mesh 2: Physical Region.....	25

LIST OF TABLES

A.1 - Sample Input for Initial Run.....	23
A.2 - Sample Input for Restart Run.....	25

(1)

ABSTRACT

INMESH is a computer program for the generation of two-dimensional boundary-fitted coordinate systems. This general program creates a numerical transformation which maps an arbitrary region in physical space onto a regular region with a uniform grid in computational space. The computational region may consist of one or more segments joined together so that any topological complexity in the physical region, such as multiple-connectedness, is suitably resolved. Numerical solutions of partial differential equations may be obtained from this computational field regardless of boundary shape. Emphasis is placed on ease of use and interactive graphics so that a mesh can be generated quickly, viewed interactively, and regenerated immediately with slight alterations until a suitable mesh configuration has been obtained. A brief explanation of the method is given along with a detailed description of the input required to produce boundary-fitted coordinate systems. Several sample meshes are shown for configurations involving foil-like bodies. *Additional Report to*

fluid dynamics; problem solution

ADMINISTRATIVE INFORMATION

This research was carried out under the Mathematical Sciences Research Program, Program Element 61153N, Task Area SR-0140301, Work Unit 1840-040, and was supported by the Naval Sea Systems Command, Code SEA 05R24.

INTRODUCTION

Research in the area of Numerical Mesh Generation has increased dramatically during the last decade. This is due in large part to rapid advances both in numerical and mathematical techniques and in the large-scale computing capabilities needed to support these techniques. The value of boundary-fitted coordinate systems as a tool for the numerical solution of partial differential equations on arbitrarily-shaped regions was illustrated in a 1974 paper by Thompson,^{1*} in which a surface-oriented grid was created about an irregularly-shaped two-dimensional "rock." A finite-difference scheme was then used to compute a simple flow solution around this body. The mesh-generating procedure consisted of numerically solving a system of elliptic equations to produce a numerical transformation that mapped the two-dimensional physical region onto a rectangular computational region which was

*A complete list of references is given on page 29.

over-laid with a uniform square grid. This mapping was defined so that boundaries of the physical region were transformed into boundaries of the computational region. The inverse image of the computational mesh is the boundary-fitted coordinate system in which each boundary is coincident with a constant coordinate line. As with the solution of the transformation, the solution of the flow problem can then take place in the computational plane (after a transformation of the governing equations) without regard to boundary shape. Thompson's study served to initiate much of the subsequent research into numerical mapping techniques.

Although the original mapping presented by Thompson was a relatively simple one, the basic procedure was applicable to many physical problems involving field solutions. In computational fluid dynamics, for example, finite-difference schemes require a grid in the region of interest so that the flow characteristics can be found at the nodal points. Difficulties often arise in such solution methods as a result of inaccurate numerical representation of boundary conditions. The boundary conditions are usually best represented when the boundaries themselves are coordinate lines so that no interpolation is necessary due to irregular cells or computational "stars." Numerically generated boundary-fitted coordinates showed promise of becoming a very valuable tool, not only in the field of computational fluid dynamics but also in the areas of heat transfer, structural mechanics, and others. Another important feature is that the method is not restricted to two dimensions as are some other widely-used mapping techniques such as complex variables.

At DTNSRDC, we are interested in flow problems involving complex geometries that often include deformable boundaries whose shape is not known in advance but is part of the solution, as in free-surface problems. Other methods had been used to simplify the computations necessary to solve these challenging problems, but boundary-fitted coordinates offered an efficient way of adapting the grid system to geometries that changed with time. In addition, all calculations are done in the computational space which remains invariant even though the shape of the physical region may change. This feature makes the method especially attractive for nonlinear water wave problems since the solution of these problems must be obtained at boundaries whose shape is not known in advance. Finite-difference schemes using boundary-fitted coordinates have been used successfully for free-surface problems involving a translating circular cylinder, a moving transom stern, and breaking wave development.^{2,3}

In applying the boundary-fitted coordinate/finite-difference method to fluid flow problems, we found that the original type of configuration, which mapped the entire physical region to a single rectangle, was not suitable for all of our geometries. To avoid rewriting the mesh generating routines for each new geometry, we needed a general program that could be used for more complex situations. A decision was made to construct a grid-generating program that could easily be applied to a wide variety of problems. The program had to be one that was not restricted to any predetermined mapping configuration, but that could be designed to accommodate even the most complicated geometric domains, including those with multiple-connectedness. This versatility can be achieved through the use of segmented computational regions tailored to handle the topological complexities of each individual problem. The proper choice of mapping configuration is particularly important for problems containing two or more boundaries of different shape, especially if these boundaries vary with time. The program also had to be modular in design so that the subroutines which linked the computational grid segments together could be used not only for the mesh generation, but also for numerical solution of the flow equations. INMESH is the result of this effort.

DESCRIPTION OF THE METHOD

The first computer program written at DTNSRDC to generate finite-difference meshes for arbitrary doubly-connected two-dimensional regions was called NUMESH and was documented in a 1977 report.⁴ NUMESH could produce only two different grid configurations. The basic method of numerical mesh generation was extended to three-dimensional regions in a 1978 report.⁵ The latter study produced some simple three-dimensional meshes about ship-like bodies. The idea of using a set of rectangular grid segments to "build" a suitable computational region for two-dimensional problems was the basis of a new program, INMESH, written in 1982.⁶ The current work gives a description of the latest version of INMESH, along with information needed for execution of the program. Several new features have been added and the input has been simplified.

The method used to map the physical region onto the segmented computational grid is based on a system of either Laplace or Poisson equations. The desired transformation from (x,y)-space to (ξ,η)-space is represented by

$$\xi = \xi(x,y) \quad \text{and} \quad \eta = \eta(x,y) \quad (1)$$

The transformations are required to be solutions of the equations

$$\xi_{xx} + \xi_{yy} = P(\xi,\eta) \quad (2)$$

and

$$\eta_{xx} + \eta_{yy} = Q(\xi,\eta)$$

subject to appropriate boundary conditions. The source functions P and Q may be zero (a Laplace generating system) or nonzero (a Poisson system) in order to influence the spacing of coordinate lines in the physical region. At present, spacing of both ξ and η lines in the physical region can be made to assume spacing similar to that of points on the boundaries. This feature, along with the proper choice of mapping configuration, should provide enough coordinate system control for most geometries.

For computational purposes, Equations (2) are transformed to (ξ,η)-space by interchanging dependent and independent variables to yield

$$\alpha x_{\xi\xi} - 2\beta x_{\xi\eta} + \gamma x_{\eta\eta} + J^2(Px_{\xi} + Qx_{\eta}) = 0 \quad (3)$$

$$\alpha y_{\xi\xi} - 2\beta y_{\xi\eta} + \gamma y_{\eta\eta} + J^2(Py_{\xi} + Qy_{\eta}) = 0$$

where

$$\begin{aligned} \alpha &= x_{\eta}^2 + y_{\eta}^2 & \beta &= x_{\xi}x_{\eta} + y_{\xi}y_{\eta} \\ \gamma &= x_{\xi}^2 + y_{\xi}^2 & J &= x_{\xi}y_{\eta} - x_{\eta}y_{\xi} \end{aligned} \quad (4)$$

Each derivative in Equations (3) and (4) is replaced by the appropriate central difference approximation, and the resulting system is solved using successive over-relaxation. A more complete discussion of this solution technique can be found in previous work.⁷

The computational region comprises rectangular grid segments which are mapped onto sections of grid in the physical region. The sides of the grid segments in computational space are joined together so as to achieve the desired distribution of corresponding grid lines in physical space. The sides of each rectangle are designated as one of two fundamental types: 1) Fixed (F-type) and 2) Matching (M-type). For the F-type side, the x- and y-coordinates of the grid points are fixed and hence do not change during the computation of the mesh. An F-type side would be found in a grid section adjacent to a boundary in physical space such as a wall or far-field boundary. M-type designations are used to connect the sides of different segments or different sides of the same segment. The two sides that are joined must both be M-type and have the same number of points. An M-type connection has an overlap of two grid lines. Unlike F-type sides, the physical coordinates of M-type sides are not determined in advance but must be computed. This property ensures the smoothness of coordinate lines throughout the physical region. The grid in the computational region must be defined so that each side of each segment is either fixed to a physical boundary or is joined to another entire side. This requirement may mean that many computational segments are needed for a complex geometry, but the input for each segment is simple. Another approach would have been to allow the joining of partial sides of segments. If the computational segments were not required to join entire side to entire side, fewer of them might be needed for a particular geometry, but the input that designated the connections would be more complicated.

USE OF THE PROGRAM

To create a coordinate system with INMESH, the user must first decide on the mapping configuration that best suits the geometry of the problem. This step often involves making a sketch of the coordinate system as it will appear in the physical plane. The grid in the physical region is then divided into sections in such a way as to ensure that each side of each segment in the computational region is either fixed to a physical boundary or is joined to another entire side. The sides are defined as one of the two types (F or M), with connections specified where appropriate.

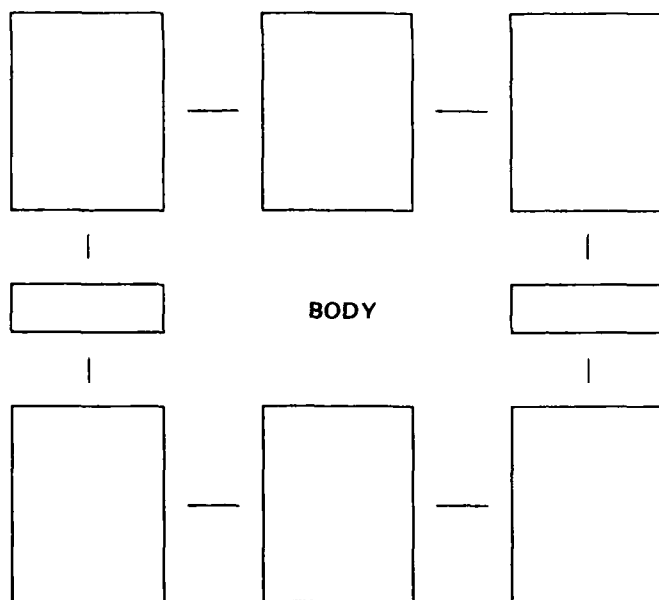
des to be used in the initialization of the mesh are set along with the coordinate control parameters for each side. Finally, the physical coordinates of the fixed sides are determined and the program is ready to be run.

All user-supplied input to the program is given in the form of 80-column card pages with variables in free format. The input consists of values of parameters which 1) define the mesh segments, 2) designate the type of each side, 3) specify sides used for initialization, 4) establish the linkage between segment sides, and provide coordinate line control. The input for each segment side is simple--two parameters are needed for connections and two for control. The physical coordinates of the fixed sides must be specified where necessary. Also included in the input are values of parameters needed for the iterative solution of the mesh. The INMESH User's Guide in Appendix A gives a complete description of the necessary input for INMESH, along with a simple example.

In addition to the printed output of INMESH, which echoes the input and gives a record of the convergence error after each iteration, two files can be created for later use. A plot file in a format suitable for use with the DTNSRDC Interactive Data Display System (IDDS) or Apollo Interactive Data Display System (AIDDS) plotting packages can be written as well as a restart file for use as input to another INMESH run or as the initial mesh for an application program.

Almost any type of boundary-fitted coordinate system can be created with INMESH. The coordinate system that is generated may be easily changed by altering the number of grid segments and connections between them in the computational region. Grid lines are added to or removed from specific areas by changing the number of points in the appropriate segments. The coordinate lines in the physical plane are controlled by varying the number of grid lines in particular segments and changing the definition of the functions P and Q in the generating equations. The flexibility provided by INMESH is especially useful for multi-body problems in which different mesh configurations are needed in different parts of the physical region.

The current FORTRAN version of INMESH is being run on an Apollo DOMAIN Scientific/Engineering Workstation (SEW). The SEW also supports interactive graphics so that a mesh can be generated quickly and viewed immediately. If changes are necessary, the mesh can be regenerated from new initial data or from the restart file produced by the previous run and modification data. This process is continued until a mesh possessing the properties necessary for a particular application is obtained. The mesh would then be transferred to a mainframe computer for subsequent



TYPICAL MESH SEGMENT

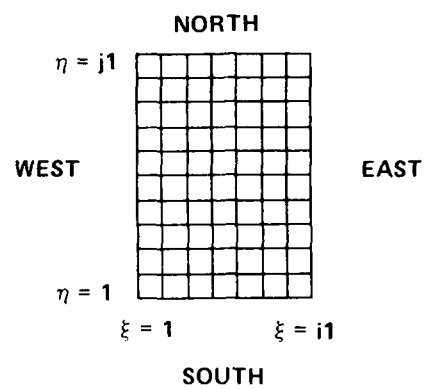


Figure A.1 - Mesh 1: Computational Region

The spacing of both ξ and η lines within an individual segment is influenced by control functions defined on each of the segment sides. These control functions are based on the boundary point distribution of fixed sides of segments in the mesh. Any fixed side may be used for the control function definition of that or any other segment side as long as the number of points agree. With this type of control, the relative spacing of the coordinate lines tends to be the same as that of the points on the controlling fixed sides. The parameters KSEG and KSID on east and west sides specify the controlling segment sides for the η lines. Control functions for the η lines in the interior of a mesh segment are then determined by linear interpolation between the control functions defined on the east and west sides. The KSEG and KSID parameters for the north and south sides control the ξ lines in a similar manner. If KSEG = KSID = 0, the control function is assumed to be based on a fixed side having a uniform distribution of points.

Type (4) input: Boundary/initialization data in the form of (x,y) pairs is required for all fixed sides. If NTYP \neq 0 in the side connection data card, MSEG = KSID = 0 and boundary data must be supplied next in the sequence. NTYP > 0 indicates that the boundary data is given in normal order, i.e. south to north for the east and west sides and west to east for the north and south sides. NTYP < 0 indicates that the boundary data is given in reverse order and is to be converted to normal order by the program before any computation is done. Fixed sides used for initialization of grid segments are designated by NTYP = ± 1 . If two opposite sides of a segment (east and west or north and south) are specified as initialization sides, linear interpolation is used to distribute points throughout the interior of the segment. Each segment with one or more fixed sides should have an initialization side or sides specified.

Example

Figure A.1 shows the segments of the computational space for a grid, Mesh 1, about a simple body located within a square outer boundary. Also shown in Figure A.1 is a typical mesh segment with sides labeled according to its orientation. Table A.1 gives the input necessary for the creation of Mesh 1 illustrated in Figure A.2. An exploded view of the physical space for this configuration is included in Figure A.2 for comparison with the corresponding computational space seen in Figure A.1.

Type (3) input: The side connection/control function parameters, in order, are as follows:

NSID, NTYP, MSEG, MSID, KSEG, KSID

<u>Parameter</u>	<u>Description</u>
NSID	Grid segment side identifier for current segment: NSID = 1, East/Right (i = i1) = 2, North/Top (j = j1) = 3, West/Left (i = 1) = 4, South/Bottom (j = 1)
NTYP	Side type identifier: NTYP = 0, Matching = +1, Fixed (initialization side) = +2, Fixed (if NTYP < 0, input is in reverse order)
MSEG	Matching segment identifier: MSEG = 0, No matching (NTYP ≠ 0) = NSEG of Matching segment (NTYP = 0)
MSID	Matching side identifier: MSID = 0, No matching (NTYP ≠ 0) = NSID of Matching side (NTYP = 0) (if MSID = -NSID, points matched in reverse order)
KSEG	Control segment identifier: KSEG = 0, Control based on equal spacing = NSEG of controlling segment
KSID	Control side identifier: KSID = 0, Control based on equal spacing = NSID of controlling side (if KSID = -NSID, points used in reverse order)

The remaining card images supply basic information for the individual grid segments.

- (2): Segment definition
- (3): Side connections/control function definition
- (4): Boundary/initialization data
- (5): Control function modification

INITIAL RUN

For an initial run (INT = 0), a sequence of card images, having the form shown in the following table, must be given.

Sequential Order	Input Type	Description
1	(1)	SOR parameters
2	(2)	Segment definition (FIRST SEGMENT)
3	(3)	Side connection/control function definition
4	(4)	Boundary data (if needed)
5	(3)	Side connection/control function definition
6	(4)	Boundary data (if needed)
7	(3)	Side connection/control function definition
8	(4)	Boundary data (if needed)
9	(3)	Side connection/control function definition
10	(4)	Boundary data (if needed)
11	(2)	Segment definition (SECOND SEGMENT)
12	(3)	Side connection/control function definition
13	(4)	Boundary data (if needed)
.	.	.
.	.	.
.	.	.

Type (2) input: The segment definition parameters, in order, are as follows:
NSEG, IL, JL

<u>Parameter</u>	<u>Description</u>
NSEG	Segment number
IL	Number of ξ (vertical) lines in current segment
JL	Number of η (horizontal) lines in current segment

APPENDIX A
INMESH USER'S GUIDE

Input needed for INMESH falls into five main categories: (1) parameters for the successive overrelaxation (SOR) scheme, (2) parameters that define the size of each segment, (3) parameters that specify the way in which the grid lines are matched and controlled, (4) physical coordinates of the boundaries, and (5) parameters to modify the control functions. On initial runs, the first four types of input are required with designated fixed sides providing an initial guess. On subsequent or restart runs, in which a previously generated mesh is used as the initial guess, only the iteration scheme parameters need be given although the control function definition may be modified. All data is provided in the form of 80-column card images with variables in free format.

Type (1) input: The first card image supplies the SOR parameters for both initial and restart runs. These parameters, in order, are as follows:

E1, E2, MAX, RF, INT

<u>Parameter</u>	<u>Description</u>
E1	Minimum $ x $ and $ y $ included in convergence test: typically E1 is about 1% of $\max(x , y)$
E2	Maximum relative change allowed for convergence: typically $0.001 \leq E2 \leq 0.1$
MAX	Maximum number of iterations allowed in one run
RF	Relaxation factor: typically $1.2 \leq RF \leq 1.7$
INT	Initialization parameter: INT = 0, Initial run = 1, Restart run

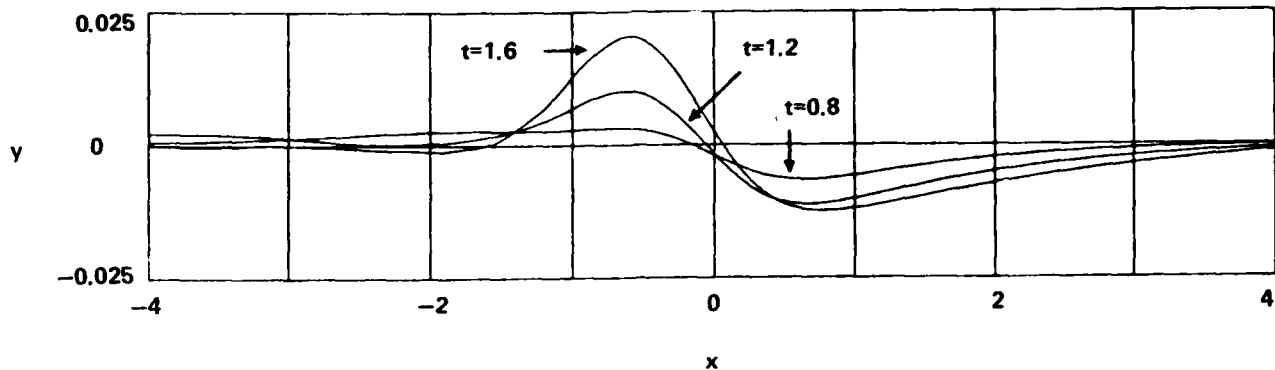


Figure 8 - Free-Surface Development for Translating Hydrofoil

CONCLUSION

A computer program (INMESH) for the generation of boundary-fitted coordinate systems has been written which makes use of segmented computational regions. This program is not restricted to certain predetermined mapping configurations and therefore can be used to produce a virtually unlimited number of different grid systems. The user may design a computational region suited to a particular geometry which makes INMESH a useful tool in the creation of curvilinear coordinate systems. The current version of INMESH represents a considerable improvement on the original program with the inclusion of coordinate line control capability. INMESH is operational on an Apollo DOMAIN Scientific/Engineering Workstation that can be used to generate and view meshes interactively. The use of INMESH was illustrated by a series of sample grids and by an application demonstrating that a segmented computational region may be used successfully in the numerical solution of a fluid dynamics problem.

ACKNOWLEDGMENT

The author wishes to thank Mr. Michael L. Brabanski for his work in developing the DPAC curve-fitting procedure for use on the Apollo SEW.

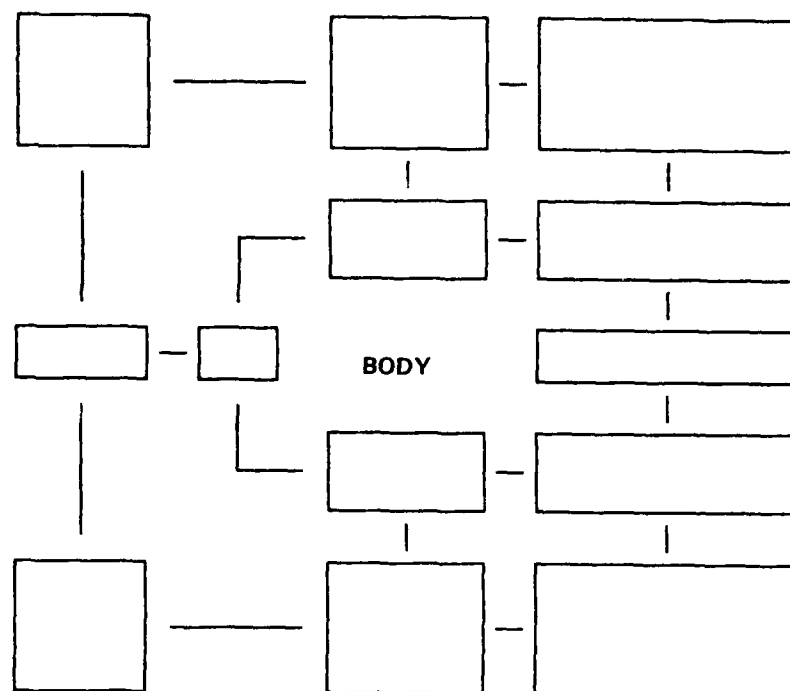


Figure 7a - Computational Region

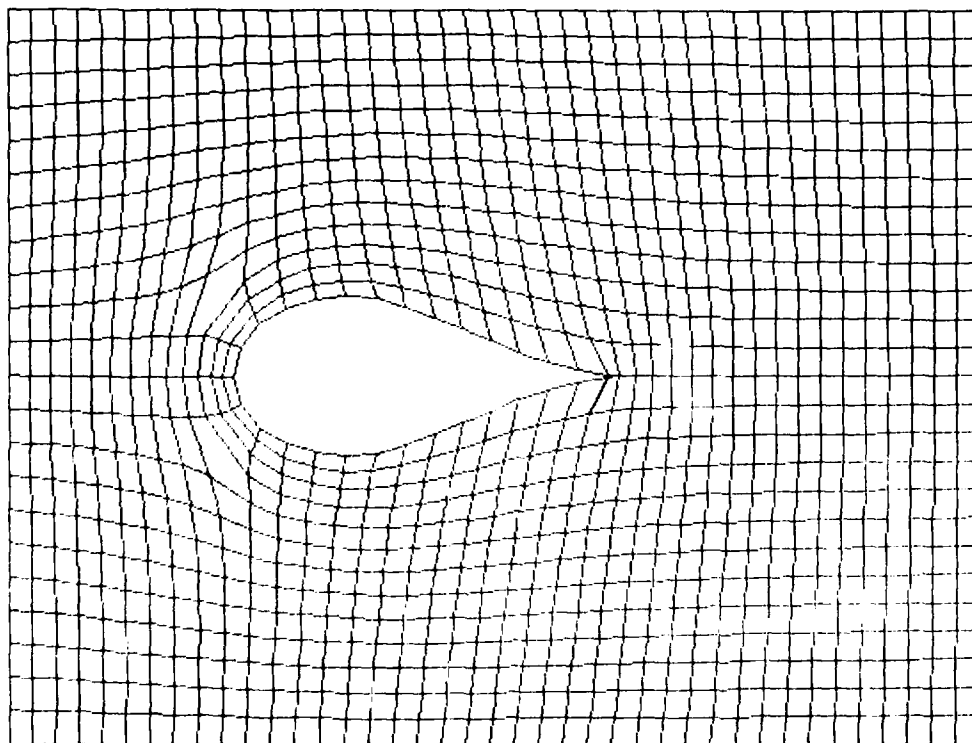


Figure 7b - Physical Region

Figure 7 - Example 4: Grid for Foil in Rectangular Region

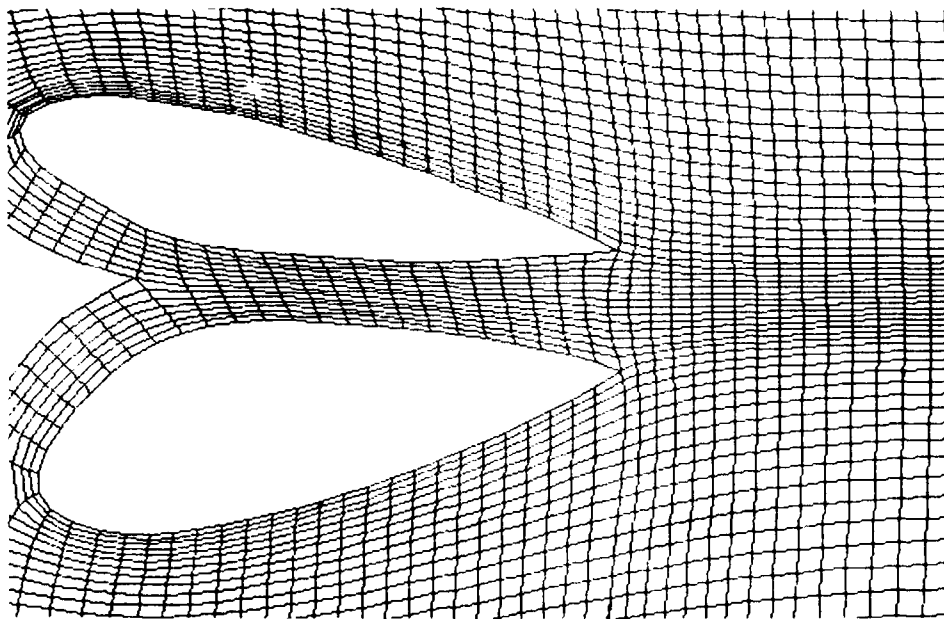


Figure 6 - Example 3: Close View of the Confluence Regions on the Multibody Foil

EXAMPLE 4

Both the inner and outer boundaries must often be considered in choosing a grid system for a particular topology. As the final example, consider a rectangular region containing a foil-like body. A useful grid for this geometry has coordinate lines wrapping around the body yet has a more rectangular nature near the outer boundary. The computational and physical regions for this mapping configuration are shown in Figure 7. In Figure 7(b), note the irregular five-sided cells that appear both above and below the foil where the grid system changes character.

A coordinate system similar to the one illustrated in Figure 7 has been used as the initial mesh for a preliminary calculation of the potential flow resulting from a hydrofoil moving beneath a free-surface. A finite-difference technique was used along with a marching scheme for the advancement of the time-dependent free-surface, requiring that a new mesh be generated at each time step as the free-surface evolves. Details of this method can be found in an earlier paper by Haussling and Coleman.⁷ Figure 8 shows the development of the free surface for various times t as the foil is abruptly put in motion. This computation was easily performed on the segmented grid since the same subroutines used to generate the mesh were also used for the flow calculations.

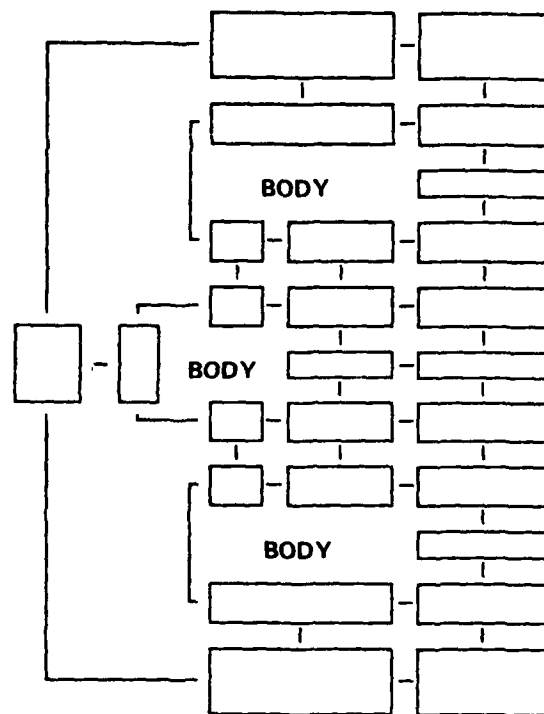


Figure 5a - Computational Region

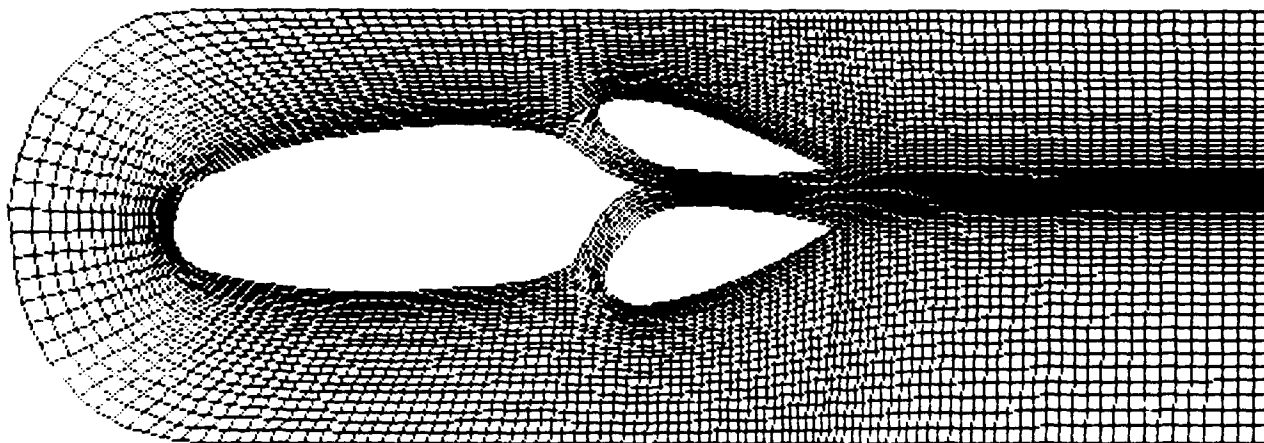


Figure 5b - Physical Region

Figure 5 - Example 3: Grid for Multibody Foil

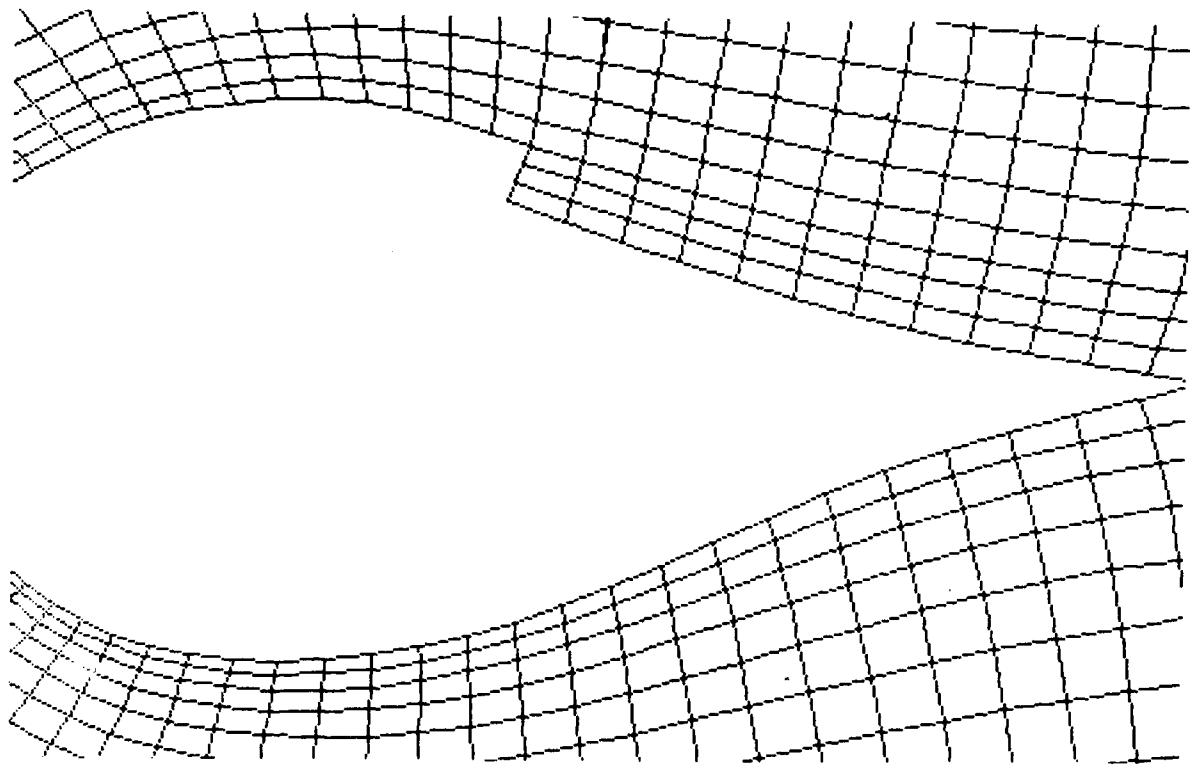


Figure 4 - Example 2: Close View of Foil with Slot

XAMPLE 3

Multibody problems can also be handled by INMESH even though the mapping configuration becomes somewhat more complicated as illustrated by Figure 5. The increase in complexity of the computational region arises from the increase in the number of discontinuous coordinate lines in the physical region. This geometry is modeled after a tri-element Augmentor-Wing airfoil being studied for use in short-takeoff-and-landing aircraft.⁸ A close-up view of the confluence region is given in Figure 6.

DPAC is given in Appendix B. To illustrate the influence of boundary-spacing control upon the mesh, another grid was generated for the same region with no coordinate line control ($P = Q = 0$). This mesh is shown in Figure 2. More information about control of the coordinate lines can be found in Appendix A.

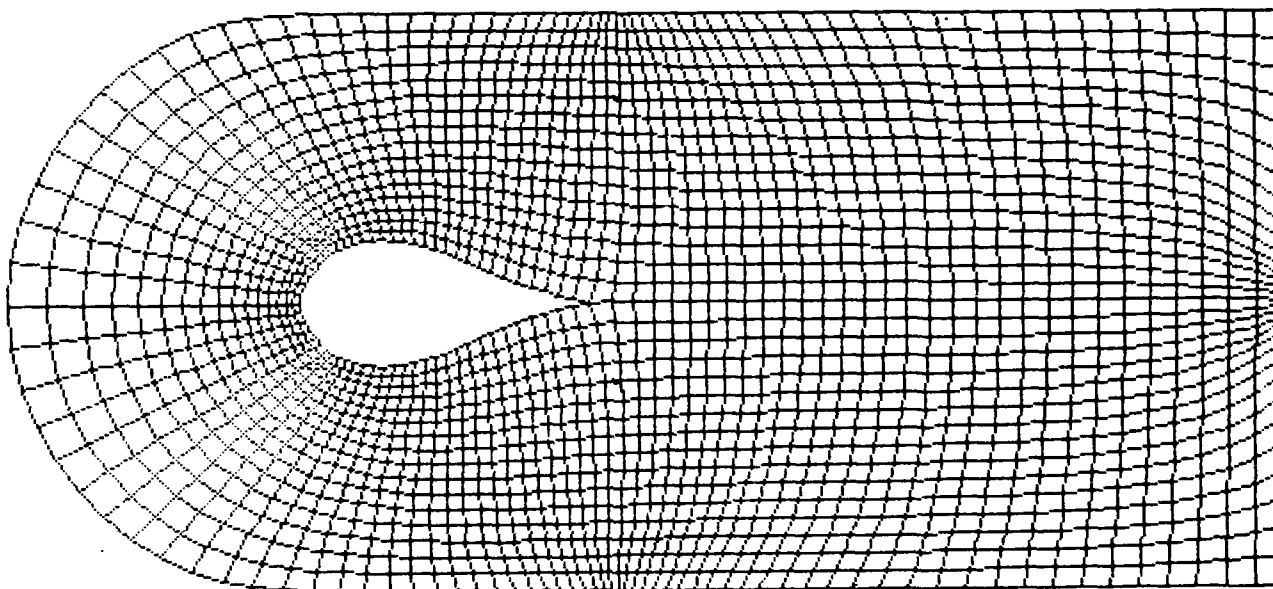


Figure 2 - Example 1: Grid For Region Shown in Figure 1 Generated without Coordinate Line Control

EXAMPLE 2

The second example illustrates a C-type grid generated about a foil that has a notch or slot in the upper surface. Figure 3 shows the computational and physical regions for this geometry. As in Example 1, the boundary-spacing control parameters were used to attract coordinate lines into the area near the body. This is a model of the cross section of an aircraft wing that has slot on the upper surface from which air is exhausted to improve the lift characteristics. Figure 4 is a close-up view of the slot.

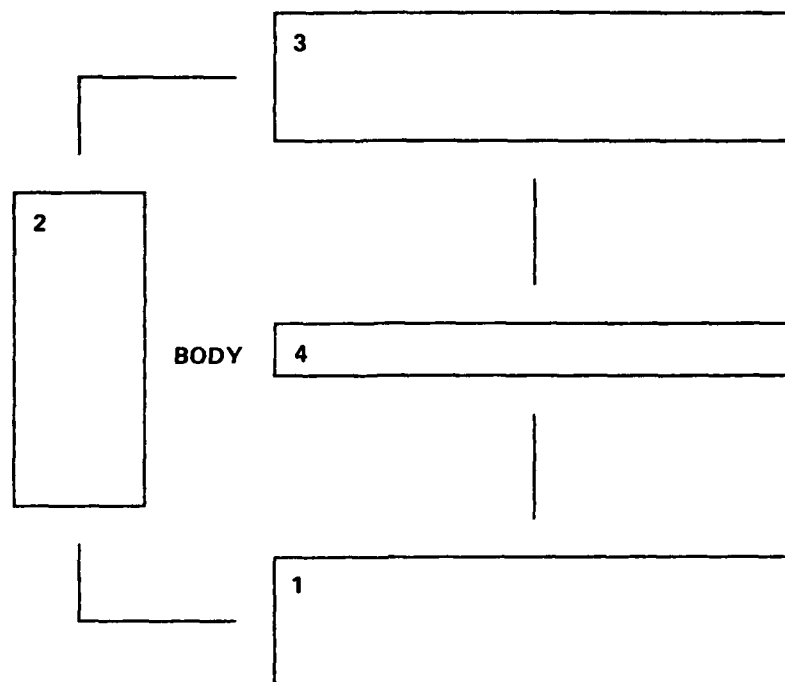


Figure 1a - Computational Region

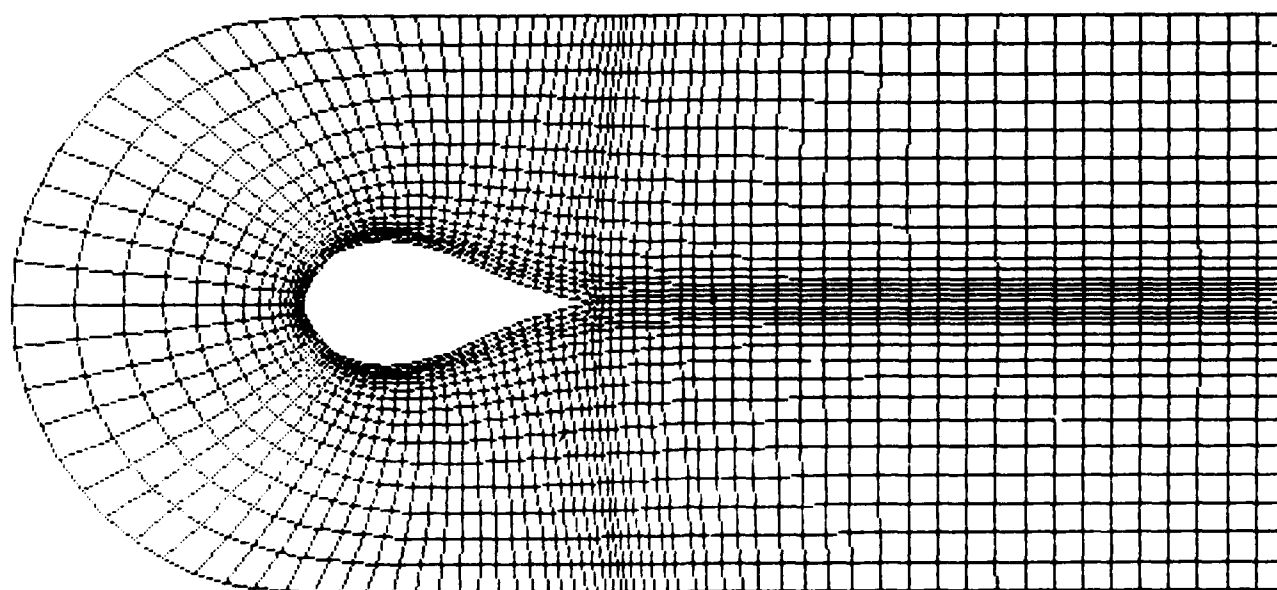


Figure 1b - Physical Region

Figure 1 - Example 1: C-type Grid for Foil

use, or perhaps another mesh, larger but with similar properties, would be produced before transfer. In this way, INMESH can aid the researcher in obtaining a "best" mesh for a particular problem by enabling him or her to efficiently produce a wide variety of grids for the same geometry.

EXAMPLES AND APPLICATIONS

EXAMPLE 1

As a first example of the meshes that can be created with INMESH, consider the 4-segment computational region of Figure 1a and the associated physical region of Figure 1b. This figure illustrates a "C-type" coordinate system about a foil-like body. In this and succeeding examples, the segmented computational region will be illustrated schematically with connections among the segments shown as curved or straight line segments. Sides with no connections given are fixed. These sketches of the computational regions show how linkage of grid segment sides determines the nature of the coordinate system as it appears in the physical region. The number of grid lines in each segment must be such that every pair of matching sides has an equal number of grid points.

Note that four computational segments are required for this example because the coordinate line that lies along the body also lies within the physical region. Segment 4 is used to link Segment 1 with Segment 3 to form a "cut" in the physical region. Although an additional mesh segment is required if a coordinate line "splits" in physical space, this approach allows all sides of all segments to be one of two basic types rather than three as in an earlier version of INMESH. This is an example of a simple mesh that has many applications.

Coordinate line control was used in this example to attract grid points into the areas around and downstream from the body. The spacing of points on the boundaries was adjusted and then the control parameters were used to maintain this spacing throughout the mesh. This particular distribution of boundary points was obtained by using a curve-fitting procedure called Distribute Points on A Curve (DPAC). DPAC runs on the Apollo SEW and uses B-splines to distribute any number of points on an arbitrary curve. The distribution can be specified explicitly or it can be generated by DPAC based on user-supplied input concerning the clustering. More detail on

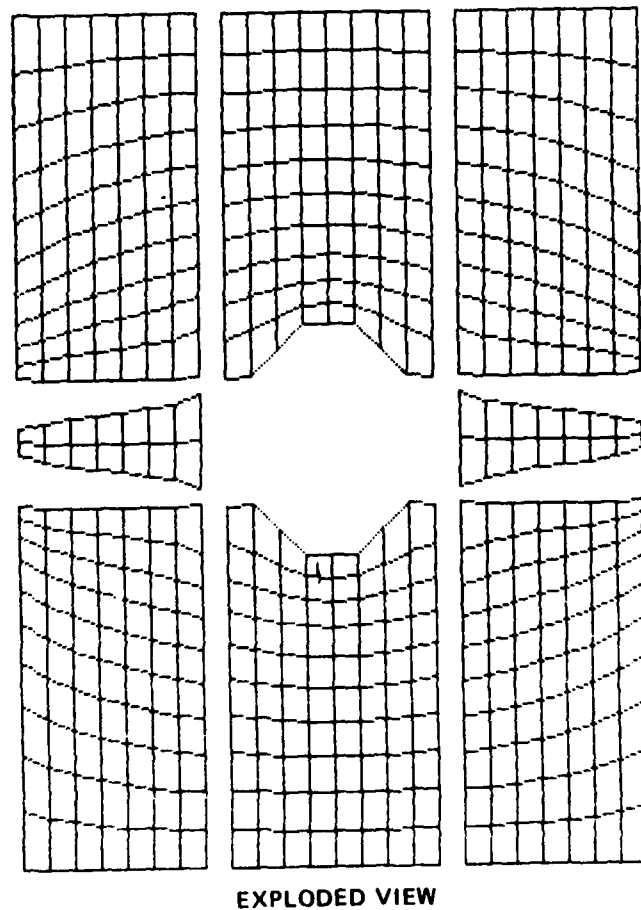
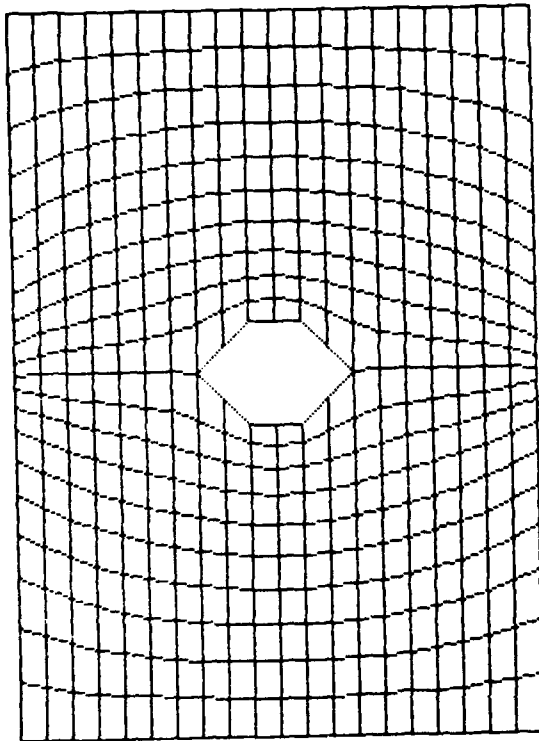


Figure A.2 - Mesh 1: Physical Region

TABLE A.1 - SAMPLE INPUT FOR INITIAL RUN

Input Type	Data	Data con't from Col. 2	Data con't from Col. 3	Data con't from Col. 4
(1)	.001 .01 100 1.5 0			
(2)	1 8 11	0.2 -1.4	3 1 0 0 0 0	-0.3 0.0
(3)	1 0 2 3 0 0	0.3 -1.4	0.3 0.0	-0.3 0.0
(3)	2 0 4 4 0 0	0.4 -1.4	0.3 0.0	-0.2 0.1
(3)	3 2 0 0 0 0	3 8 11	0.3 0.0	-0.1 0.2
(4)	-1.0 -1.40	1 2 0 0 0 0	4 0 3 2 0 0	0.0 0.2
	-1.0 -1.17	1.0 -1.40		0.1 0.2
	-1.0 -0.96	1.0 -1.17	6 8 11	0.2 0.1
	-1.0 -0.77	1.0 -0.96	1 0 7 3 0 0	0.3 0.0
	-1.0 -0.60	1.0 -0.77	2 1 0 0 0 0	0.3 0.0
	-1.0 -0.45	1.0 -0.60	-1.0 1.4	8 8 11
	-1.0 -0.32	1.0 -0.45	-0.9 1.4	1 2 0 0 0 0
	-1.0 -0.21	1.0 -0.32	-0.8 1.4	1.0 0.0
	-1.0 -0.12	1.0 -0.21	-0.7 1.4	1.0 0.05
	-1.0 -0.05	1.0 -0.12	-0.6 1.4	1.0 0.12
	-1.0 0.0	1.0 -0.05	-0.5 1.4	1.0 0.21
(3)	4 1 0 0 0 0	1.0 1.0	-0.4 1.4	1.0 0.32
(4)	-1.0 -1.4	2 0 5 4 0 0	-0.3 1.4	1.0 0.45
	-0.9 -1.4	3 0 2 1 0 0	3 2 0 0 0 0	1.0 0.60
	-0.8 -1.4	4 1 0 0 0 0	-1.0 0.0	1.0 0.77
	-0.7 -1.4	0.3 -1.4	-1.0 0.05	1.0 0.96
	-0.6 -1.4	0.4 -1.4	-1.0 0.12	1.0 1.17
	-0.5 -1.4	0.5 -1.4	-1.0 0.21	1.0 1.40
	-0.4 -1.4	0.6 -1.4	-1.0 0.32	2 1 0 0 0 0
	-0.3 -1.4	0.7 -1.4	-1.0 0.45	0.3 1.4
(2)	2 9 11	0.8 -1.4	-1.0 0.60	0.4 1.4
(3)	1 0 3 3 0 0	0.9 -1.4	-1.0 0.77	0.5 1.4
(3)	2 2 0 0 0 0	1.0 -1.4	-1.0 0.96	0.6 1.4
(4)	-0.3 0.0	4 8 3	-1.0 1.17	0.7 1.4
	-0.3 0.0	1 1 0 0 0 0	-1.0 1.40	0.8 1.4
	-0.2 -0.1	-0.3 0.0	4 0 4 0 0 0	0.9 1.4
	-0.1 -0.2	-0.3 0.0	7 9 11	1.0 1.4
	0.0 -0.2	-0.3 0.0	1 0 8 3 0 0	3 0 7 1 0 0
	0.1 -0.2	2 0 6 4 0 0	2 1 0 0 0 0	4 0 5 2 0 0
	0.2 -0.1	3 2 0 0 0 0	-0.4 1.4	
	0.3 0.0	-1.0 -0.1	-0.3 1.4	
	0.3 0.0	-1.0 0.0	-0.2 1.4	
(3)	3 0 1 1 0 0	-1.0 0.1	-0.1 1.4	
(3)	4 1 0 0 0 0	4 0 1 2 0 0	0.0 1.4	
(4)	-0.4 -1.4	5 8 3	0.1 1.4	
	-0.3 -1.4	1 2 0 0 0 0	0.2 1.4	
	-0.2 -1.4	1.0 -0.1	0.3 1.4	
	-0.1 -1.4	1.0 0.0	0.4 1.4	
	0.0 -1.4	1.0 0.1	3 0 6 1 0 0	
	0.1 -1.4	2 0 8 4 0 0	4 2 0 0 0 0	

RESTART RUN

For a restart run (INT = 1), only the SOR parameter card (1) is required. The user also has the option of altering the control functions of the previously generated mesh. This is done by supplying one control function modification card for each segment side on which the function is being redefined. Thus the complete sequence of card images needed for a restart run is as follows:

- (1) SOR parameters
- (5) Control function modification (if needed)
- (5) Control function modification (if needed)
- :
- etc.
- :

Type (5) input: The control function modification parameters, in order, are as follows:

NSEG, NSID, KSEG, KSID

<u>Parameter</u>	<u>Description</u>
NSEG	Segment for which control is being modified
NSID	Side on which control is being modified
KSEG	Controlling segment identifier
KSID	Controlling side identifier

Example

Input data used for a restart run with control function modification is shown in Table A.2. The output for Mesh 1 was used as a basis for this restart, which produced Mesh 2 (as shown in Figure A.3). A comparison of Mesh 1 and Mesh 2 shows how the control parameters can affect the curvature of the coordinate lines in the physical space.

TABLE A.2 - SAMPLE INPUT FOR RESTART RUN

Input Type	Data	Data con't from Col. 2	Data con't from Col. 3
(1)	.001 .001 100 1.5 1		
(5)	1 1 1 3 1 3 1 3 2 1 3 1 2 3 1 3	3 1 3 1 3 3 3 1 6 1 6 3 6 3 6 3	7 1 8 1 7 3 6 3 8 1 8 1 8 3 8 1

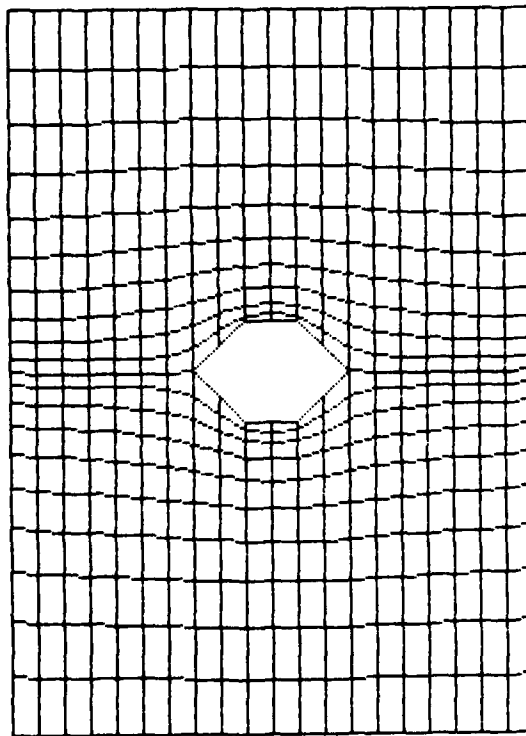


Figure A.3 - Mesh 2: Physical Region

APPENDIX B

DPAC: DISTRIBUTE POINTS ON A CURVE

INMESH uses an elliptic system of partial differential equations (a Laplace or Poisson system) to generate numerical grids. The curvilinear coordinates, ξ and η , are specified on the boundaries from which the Cartesian coordinates, x and y , are generated in the bounded region. The boundary value problem is then solved on the physical field with boundary conditions specified on curved boundaries. Correct specification of these boundary conditions is thus crucial to obtaining the desired mesh. DPAC is a set of numerical procedures designed to aid the user in producing a distribution of boundary points so that a suitable coordinate system results. While the interactive curve-fitting capability provided by DPAC is not necessary for grid generation with INMESH, it is highly desirable. The ability to change the boundary point distribution quickly and easily becomes even more important as the complexity of the coordinate system increases.

DPAC was originally written for use with the IDDS plotting package on the CDC 176 at DTNSRDC. The procedures have been converted to run on the Apollo DOMAIN Scientific/Engineering Workstation in conjunction with the AIDDS plotting system. To the user, DPAC appears as a series of interactive commands. After the user enters a command with the appropriate parameters, the corresponding procedure is executed and the results are displayed graphically. The results, which have been written to a specified file, can then be accepted or else additional commands can be entered until satisfactory results are obtained.

A report currently in preparation will describe the use of DPAC in detail. The following is a brief summary of the basic DPAC commands used to produce the boundary point distribution for the examples in this report.*

DEF [nfile] [nb]

The DEF command produces a B-spline fit using "nb" basis functions to the curve described in discrete form on file "nfile." The nb parameter determines the "stiffness" of the fit. The original points are displayed along with the B-spline curve that has been fit to them.

*For more information on DPAC, contact Mr. M. L. Brabanski, DTNSRDC, Code 1843.

EVEN [np]

The EVEN command distributes "np" equally spaced points on the curve defined by DEF. Both the points and the curve are displayed. Also shown in the display is the interval [0,1] with an equivalent distribution of points.

EVALF [nfile]

The EVALF command distributes points on a curve defined by DEF in the same manner as a distribution on [0,1] specified in "nfile." The display shows the distribution on both the curve and [0,1].

ZONES [nfile] [np]

The ZONES command aids the user in defining a distribution if the desired distribution is not known exactly. A distribution of "np" points on [0,1] is created by specifying in "nfile" subintervals of [0,1] where the spacing between points is expanding, contracting, or remaining constant. The distribution thus defined is then automatically transferred to the curve by EVALF. Both distributions, on the curve and on [0,1], are displayed.

REFERENCES

1. Thompson, J.F. et al., "Automatic Numerical Generation of Body-Fitted Curvilinear Coordinate System for Field Containing Any Number of Arbitrary Two-Dimensional Bodies," J. Comp. Phys., Vol 15, pp. 299-319 (1974).
2. Haussling, H.J. and R.M. Coleman, "Nonlinear Water Waves Generated by an Accelerated Circular Cylinder," J. Fluid Mech., Vol. 92, Part 4, pp. 767-781 (1979).
3. Coleman, R.M. and H.J. Haussling, "Nonlinear Waves Behind and Accelerated Transom Stern," Proc. 3rd Int. Conf. Numerical Ship Hydrodyn., Palais des Congres, Paris, France, pp. 111-118 (1981).
4. Coleman, R.M., "NUMESH: A Computer Program to Generate Finite-Difference Meshes for Arbitrary Doubly-Connected Two-Dimensional Regions," Report DTNSRDC/CMLD-77-05 (Mar 1977).
5. Coleman, R.M., "Boundary-Fitted Coordinate Systems for Three-Dimensional Regions Containing Ship-Like Bodies," Report DTNSRDC-78/085 (Sept 1978).
6. Coleman, R.M., "Generation of Boundary-Fitted Coordinate Systems Using Segmented Computational Regions," Numerical Grid Generation, J.F. Thompson, ed., Elsevier Science Publishing Co., Inc., N.Y. (1982).
7. Haussling, H.J. and R.M. Coleman, "Finite-Difference Computations Using Boundary-Fitted Coordinates for Free-Surface Potential Flows Generated by Submerged Bodies," Proc., 2nd Int. Conf. Numerical Ship Hydrodyn., Univ. of California at Berkeley, pp. 221-233 (1977).
8. Sorenson, R.L., "Grid Generation by Elliptic Partial Differential Equations for a Tri-Element Augmentor-Wing Airfoil," Numerical Grid Generation, J.F. Thompson, ed., Elsevier Science Publishing Co., Inc., N.Y. (1982).

INITIAL DISTRIBUTION

Copies

5 CONR
 1 400 Saalfeld
 1 411 Wegman
 1 411MA Holland
 2 432 Whitehead
 Lee

1 NRL
 1 5840 Skop

1 NAVMAT
 1 07

4 NAVSEA
 1 05R24
 1 55W3 Comstock
 1 55W33 Sandburg
 Chen

3 NAVPGSCOL
 1 Sarpkaya
 1 Math Dept
 1 Library

1 NAVWARCOL

1 NSWC/Dahlgren/Lib

2 NSWC/White Oak
 2 R44 Solomon
 Zein

1 NAVSHIPYD BREM/Lib

1 NAVSHIPYD CHASN/Lib

1 NAVSHIPYD MARE/Lib

1 NAVSHIPYD NORVA/Lib

1 NAVSHIPYD PEARL/Lib

1 NAVSHIPYD PHILA/Lib

1 NAVSHIPYD PTSMH/Lib

Copies

12 DTIC

1 BUSTAND/Lib

1 NASA HQS/Lib

4 NASA Ames Res Cen
 1 Martin
 1 MacCormick
 1 Sorenson
 1 Lib

1 NASA Lewis Res Cen/Lib

1 NASA Marshall SFC/Lib

1 J. F. Thompson
 Dept of Aerospace Eng.
 Drawer A
 Mississippi State, MS 39762

1 U. Ghia
 Dept of Aerospace Eng.
 and Applied Mechanics
 Univ. of Cincinnati
 Cincinnati, OH 45221

1 B.J. McCartin
 Pratt & Whitney Aircraft
 Mail Stop EB2C
 400 Main Street
 East Hartford, CT 06108

1 J. Dannenhoffer
 Pratt & Whitney Aircraft
 Mail Stop EB2H
 400 Main Street
 East Hartford, CT 06108

1 A.L. Loeffler, Jr.
 R & D Center
 Mail Stop A08/35
 Grumman Aerospace Corp.
 Bethpage, NY 11714

Copies

- 1 A.S. Dvinsky
Creare R & D Inc.
Great Hollow Road
P.O. Box 71
Hanover, NH 03755
- 1 R.E. Smith
NASA Langley Research Center
Hampton, VA 23665
- 1 W.J. Gordon
Dept of Mathematical Sciences
Drexel University
Philadelphia, PA 19104
- 1 P.R. Eiseman
Dept of Applied Physics and
Nuclear Engineering
Columbia University
New York, NY 10027
- 1 P.E. Rubbert
Boeing Airplane Company
Seattle, WA 98124
- 1 J.U. Brackbill
Applied Theoretical Division
Los Alamos National Laboratory
Los Alamos, NM 87545
- 1 J.L. Steger
Dept of Aeronautics
Stanford University
Stanford, CA 94305
- 1 D.A. Anderson
Dept of Aerospace Engineering
Iowa State University
Ames, IA 50011
- 1 H.A. Dwyer
Mechanical Engineering Dept
University of California
Davis, CA 95616

Copies

- 1 D.D. Knight
Dept of Mechanical and
Aerospace Engineering
Rutgers University
New Brunswick, NJ 08903
- 1 B.H. Johnson
U.S. Army Engineers Waterways
Experiment Station
P.O. Box 631
Vicksburg, MS 39180
- 1 A.J. Baker
Dept of Engineering Science
Perkins Hall
University of Tennessee
Knoxville, TN 37916
- 1 W-H. Jou
Flow Research Company
21414 68th Avenue South
Kent, WA 98031
- 1 D. Halsey
Aerodynamics Research Dept
Douglas Aircraft Company
3855 Lakewood Blvd.
Long Beach, CA 90846
- 1 B.C-J. Chen
Argonne National Laboratory
9700 South Cass Avenue
Argonne, IL 60439
- 1 P.D. Thomas
Applied Mechanics Laboratory
Lockheed Research Laboratory
3251 Hanover St.
Palo Alto, CA 94304

Copies		Copies	Code	Name
1	C.F. Shieh	1	1843	H.J. Haussling
	Calspan Field Services, Inc.	30	1843	R.M. Coleman
	AEDC Division	1	1844	S.K. Dhir
	Arnold Air Force Station, TN 37389			
1	G.D. Kerlick	1	185	T. Corin
	Nielsen Engineering & Research	1	189	G. Gray
	510 Clyde Avenue			
	Mountain View, CA 94043			
1	J.W. Thomas	1	19	M.M. Sevik
	Dept of Mathematics	1	272	E. Quandt
	Colorado State University	1	2721	R.K. Muench
	Ft. Collins, CO 80523			

CENTER DISTRIBUTION

Copies	Code	Name	Copies	Code	Name
1	01	A. Powell	1	274	L. Agiro
1	012	E. O'Neill	1	283	H. Singerman
1	012.3	D. Moran	1	2862	P.D. Conroy
1	012.4	R. Stevens	10	5211.1	Reports Distribution
1	12	W. Dietz	1	522.1	TIC (C)
1	15	W.B. Morgan	1	522.2	TIC (A)
1	1542	T. Huang			
		F. Noblesse			
1	1544	C. Dai			
1	16	H. Chaplin			
2	1606	T.C. Tai			
		H. Berman			
1	17	W.W. Murray			
2	1808s				
1	182	A.W. Camara			
1	184	J.W. Schot			

END

FILMED

11-85

DTIC